TABLE NO | | | |

STUDENT ID NO | | | | | | | | | | |

MULTIMEDIA UNIVERSITY

SUBJECT CODE _____

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 2, 2017/2018

TSE3351 – SOFTWARE EVOLUTION AND MAINTENANCE

(All sections / Groups)

8 MARCH 2018
2:30 pm – 4:30 pm
(2 Hours)

Question	Mark
A	
B	
C	
D	
Total	

Examiner 1 Signature: _____

Examiner 2 Signature: _____

Examiner 3 Signature: _____

INSTRUCTIONS TO STUDENTS

1. This question paper consists of 10 printed pages (including cover page) with 4 Sections only.

2. Attempt **ALL** questions in **SECTION A, SECTION B, SECTION C and SECTION D.** The distribution of the marks for each question is given.

3. Please write all your answers **CLEARLY** in the specific answer box provided for each question. Submit this question paper at the end of the examination.

Attempt ALL questions in SECTIONS A, B, C and D.

Section A (12.5 marks)

A good software maintenance framework will consider SPE classification scheme and eight Lehman laws of evolution. SPE refers to S (Specified), P (Problem), and E (Evolving) programs.

Based on the above context, answer questions Q-A1 to Q-A3:

A1. Explain why the SPE classification scheme is useful in software evolution. The SPE classification scheme can be categorized into TWO categories. List the TWO categories involving this SPE classification scheme.

(1 + 2 marks)

A2a. The free and open source software (FOSS) system follows a software development philosophy or rule that highlights the importance of release management in open source. What is the philosophy or rule that an FOSS system has to follow?

(0.5 + 0.5 mark)

Continued...

A2b. What are the THREE Lehman laws that are not suitable for a large-scale FOSS system such as Linux? For each of the law identified, explain why the particular law is not suitable for a large-scale FOSS system.

(1.5 + 1.5 + 1.5 marks)

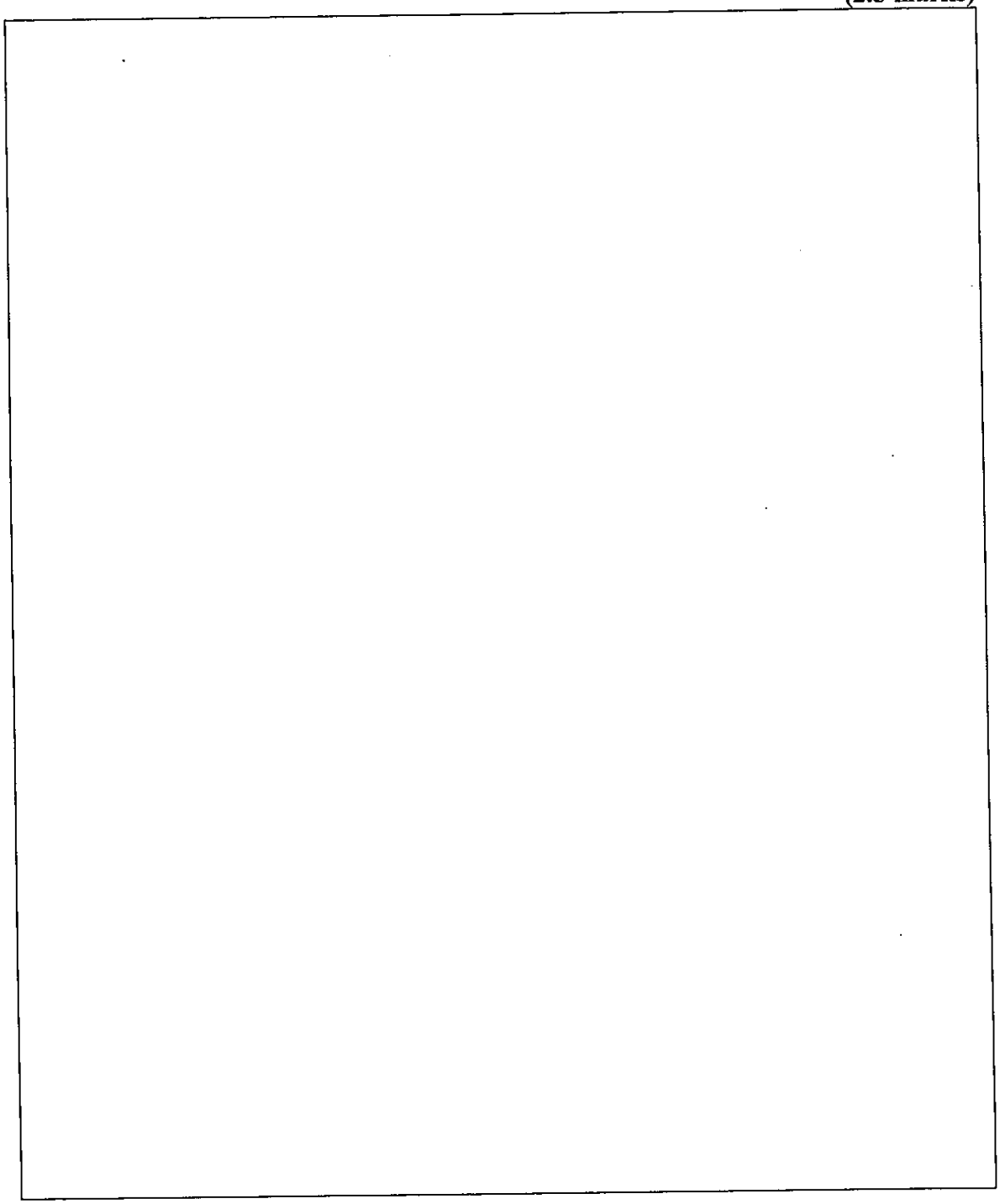
A2c. List THREE examples of small-scale FOSS systems that follow Lehman laws for software evolution.

(1.5 marks)

Continued...

A3. An E-type (Evolving) program is one that is embedded in the real world and it changes as the world does. The E-type framework consists of EIGHT components: *Formal statement of problem*, *Abstraction*, *Requirements specification*, *Program*, *Solution*, *Change*, *Compare*, and *Real world*. Draw an E-type diagram showing the relationship between the EIGHT components.

(2.5 marks)



Continued...

Section B (12.5 marks)

Consider a software system using flying unmanned aerial vehicles (UAVs) to deliver commercial goods to consumers. We expect the UAVs to behave as described in their specifications. For example, we assume that "The Model-X drone is able to transmit messages at distances up to 1,000 meters," as specified by its manufacturer. So, we write the safety specification, "The UAV must remain within 900 meters of a ground control station at all times." This minimizes the likelihood of accidents associated with the lack of direct UAV control.

Based on the above scenario, answer questions Q-B1 to Q-B5:

B1. Draw the generic life cycle of software production process for the abovementioned UAV system.

(3 marks)

B2. What are the THREE components of a maintenance process when maintaining the UAV system? Why do we need to perform the UAV software production process?

(1.5 + 0.5 marks)

Continued...

B3. You are appointed as the software maintainer of this system. Describe the FOUR categories of change that a software maintainer needs to carry out.

(4 marks)

B4. Code-and-Fix model is a simple 2-phases model. Explain the TWO weaknesses of Code-and-Fix model when applied to the above UAV system.

(1 + 1 marks)

B5. Why do we still use Code-and-Fix model for the UAV system knowing that there are weaknesses in this model? How do we use Code-and-Fix model with other larger model?

(1 + 0.5 marks)

Continued...

Section C (12.5 marks)

Building Information Modeling Security (BIM-Sec) system is a software tool that support the access control of cyber-physical spaces, such as smart buildings. Using this system, security analysts can use to visualize and edit a building's topological characteristics and verify whether access-control policies satisfy the security requirements imposed on reachability relationships—for example, to verify whether an individual can reach a specific asset or building area. If verification fails, the tool provides guidance on how to revise access-control policies for the current topological configuration. The representation of a building's topological characteristics is tailored to meet security analysts' needs with a representation of a building cyberspace, omitting those details of the physical building that are likely to be irrelevant for security analysts.

Based on the above scenario, answer questions Q-C1 to Q-C3:

C1. RationaleNode and ModelElement classes are commonly found in a typical functionality implementation. Discuss how developers can make use of these classes to develop the abovementioned BIM-Sec software system. Subsequently, draw a class diagram to support your answer.

(3 + 3 marks)

Continued...

C2. Explain how the Source Code Control System (SCCS) can be used as part of software maintenance for the BIM-Sec software system.

(2 marks)

C3. Based on Question C2, explain how changing software causes problems that require revision control in relation to SCCS use in a parallel development/maintenance environment, and explain how these problems can be overcome in other support tools.

(4.5 marks)

Continued .....

Section D (12.5 marks)

Consider an autonomous driving system, which demands highly interactive services with multi-sensor capability — far from the currently deployed functionally isolated control units. Note that 'autonomous' means the ability of a car to perform driving and make decisions without help from anyone else. Connectivity within this system is transforming cars into distributed IT (Information Technology) systems with cloud access; over-the-air (OTA) functional upgrades; and high-bandwidth access to map services, other vehicles, and the surrounding infrastructures.

D1. There are SIX variables that we can use to measure the above software system. List and explain the SIX variables with examples.

(6 marks)

Continued...

D2. What are the THREE guidelines for selecting the above software maintenance measures?

(3 marks)

D3. Maintainability refers to the ease with which program can be corrected, adapted and/or enhanced. Why does maintainability considered as an external attribute?

(1 mark)

D4. Why does configuration management in software development differ from configuration management in software maintenance?

(2.5 marks)

End of Paper